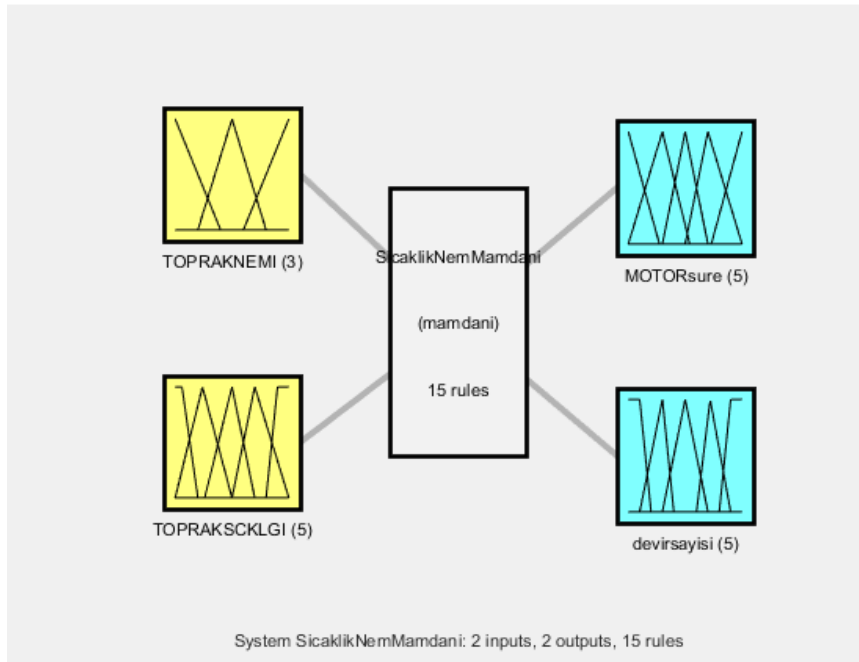




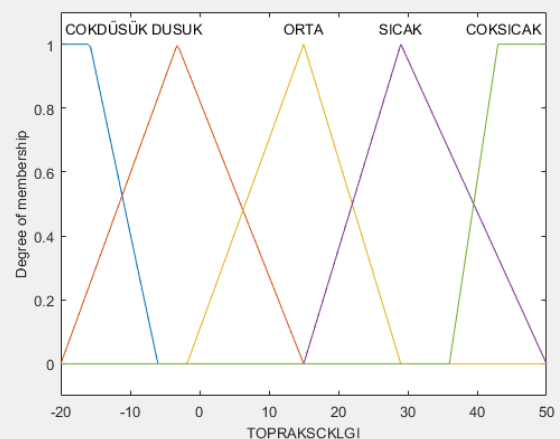
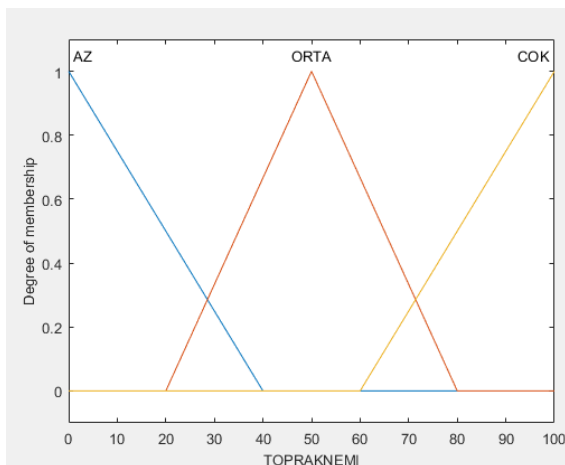
BULANIK MANTIK ile KONTROL DERS-4 YARDIMCI NOTLARI -2019

1. Birden fazla çıkışı olan Bulanık Mantık Çıkarsama Sistemleri

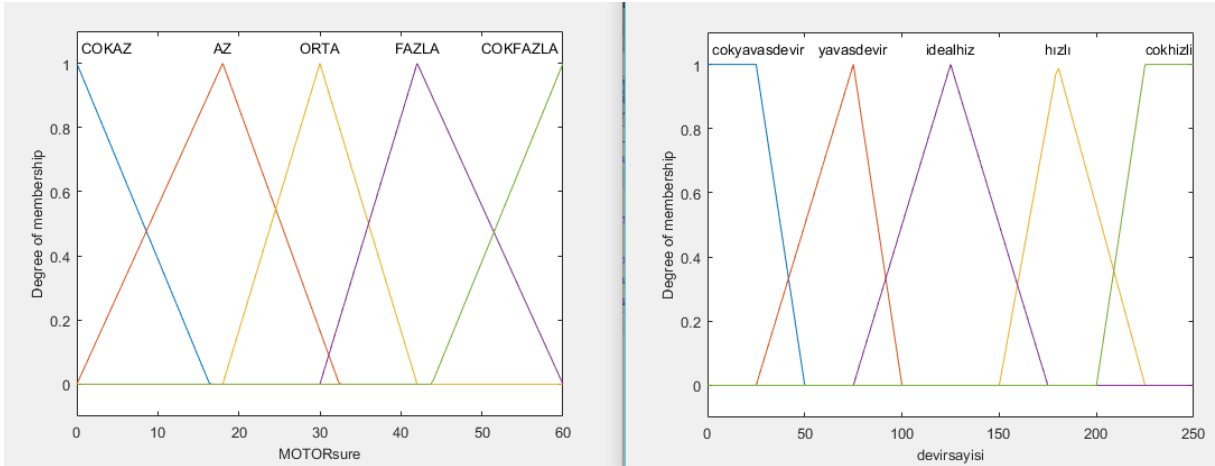
Bulanık sistemlerde tek çıkış olması gerekmez. Birden fazla çıkış da olabilir. Örnek vermek gerekirse:



Girişler:



Çıkışlar:



Kurallar:

1. If (TOPRAKNEMI is AZ) and (TOPRAKSCKLGI is COKDÜSÜK) then (MOTORsure is COKFAZLA)(devirsayisi is hizli) (1)
2. If (TOPRAKNEMI is ORTA) and (TOPRAKSCKLGI is COKDÜSÜK) then (MOTORsure is COKFAZLA)(devirsayisi is hizli) (1)
3. If (TOPRAKNEMI is COK) and (TOPRAKSCKLGI is COKDÜSÜK) then (MOTORsure is FAZLA)(devirsayisi is idealhiz) (1)
4. If (TOPRAKNEMI is AZ) and (TOPRAKSCKLGI is DUSUK) then (MOTORsure is FAZLA)(devirsayisi is idealhiz) (1)
5. If (TOPRAKNEMI is ORTA) and (TOPRAKSCKLGI is DUSUK) then (MOTORsure is FAZLA)(devirsayisi is hizli) (1)
6. If (TOPRAKNEMI is COK) and (TOPRAKSCKLGI is DUSUK) then (MOTORsure is ORTA)(devirsayisi is yavasdevir) (1)
7. If (TOPRAKNEMI is AZ) and (TOPRAKSCKLGI is ORTA) then (MOTORsure is ORTA)(devirsayisi is hizli) (1)
8. If (TOPRAKNEMI is ORTA) and (TOPRAKSCKLGI is ORTA) then (MOTORsure is ORTA)(devirsayisi is yavasdevir) (1)
9. If (TOPRAKNEMI is COK) and (TOPRAKSCKLGI is ORTA) then (MOTORsure is ORTA)(devirsayisi is yavasdevir) (1)
10. If (TOPRAKNEMI is AZ) and (TOPRAKSCKLGI is SICAK) then (MOTORsure is ORTA)(devirsayisi is hizli) (1)
11. If (TOPRAKNEMI is ORTA) and (TOPRAKSCKLGI is SICAK) then (MOTORsure is AZ)(devirsayisi is yavasdevir) (1)
12. If (TOPRAKNEMI is COK) and (TOPRAKSCKLGI is SICAK) then (MOTORsure is AZ)(devirsayisi is yavasdevir) (1)

13. If (TOPRAKNEMI is AZ) and (TOPRAKSCKLGI is COKSICAK) then (MOTORsure is ORTA)(devirsayisi is yavasdevir) (1)

14. If (TOPRAKNEMI is ORTA) and (TOPRAKSCKLGI is COKSICAK) then (MOTORsure is AZ)(devirsayisi is yavasdevir) (1)

15. If (TOPRAKNEMI is COK) and (TOPRAKSCKLGI is COKSICAK) then (MOTORsure is COKAZ)(devirsayisi is cokyavasdevir) (1)

2. Komut satırından Bulanık Mantık Çıkarsama Sistemi oluşturulması

Fuzzy araç kutusu kullanarak çıkarsama sistemi oluşturmak bazı yönlerden avantajlıdır. Ancak dinamik sistemler oluşturmak için ve kurduğumuz sistemin parametrelerini hızlıca değiştirip yeni denemeler yapabilmek için komut satırından girebileceğimiz kodlara ihtiyaç duyulacaktır.

Örneğin; araç kutusunda oluşturduğumuz sistemi her bir giriş için her seferinde tek bir kesin (crisp) değer ile deneyebiliriz. Ancak bu yeterli değildir. Çoğu zaman sistemi bir dizi ya da matris içinden gelen çoklu veriler ile art arda test etmemiz ve hataları hızlıca düzeltmemiz gerekecektir. Bu durumda kod satırından ya da m. files üzerinden değişiklikleri yapmak en iyi seçenek olacaktır.

Risk problemi için kodlar aşağıdaki gibi olacaktır. Kodlar bir m. file içerisinde yazılmıştır.

Yöntem 1: Kod yapısı açık ve anlaşılır bir yapıdadır. Ancak kod satır sayısı fazladır.

```
clear;
clc;
a = newfis('riskAnaliz2');
a.type='mamdani';
a.input(1).name = 'ProjeButce';
a.input(1).range = [0 100];
a.input(1).mf(1).name = 'yetersiz';
a.input(1).mf(1).type = 'trapmf';
a.input(1).mf(1).params = [-36 -4 25 47];
a.input(1).mf(2).name = 'sikisik';
a.input(1).mf(2).type = 'trimf';
a.input(1).mf(2).params = [28 52 76];
a.input(1).mf(3).name = 'yeterli';
a.input(1).mf(3).type = 'trapmf';
a.input(1).mf(3).params = [60 80 104 104];
a.input(2).name = 'CalisanSayisi';
a.input(2).range = [0 100];
a.input(2).mf(1).name = 'az';
a.input(2).mf(1).type = 'trapmf';
a.input(2).mf(1).params = [-36 -4 30 65];
a.input(2).mf(2).name = 'fazla';
a.input(2).mf(2).type = 'trapmf';
a.input(2).mf(2).params = [35 69 100 100];
a.output(1).name = 'risk';
a.output(1).range = [0 100];
a.output(1).mf(1).name = 'dusuk';
a.output(1).mf(1).type = 'trapmf';
a.output(1).mf(1).params = [-36 -4 22 39];
a.output(1).mf(2).name = 'normal';
a.output(1).mf(2).type = 'trapmf';
a.output(1).mf(2).params = [30 46 54 70];
a.output(1).mf(3).name = 'yukseki';
```

```
a.output(1).mf(3).type = 'trapmf';
a.output(1).mf(3).params = [61 78 100 100];
%antecedent öncül, consequent ardıcıl
a.rule(1).antecedent = [3 1];
a.rule(1).consequent = [1];
a.rule(1).weight = 1;
a.rule(1).connection = 2;
a.rule(2).antecedent = [2 2];
a.rule(2).consequent = [2];
a.rule(2).weight = 1;
a.rule(2).connection = 1;
a.rule(3).antecedent = [1 0];
a.rule(3).consequent = [3];
a.rule(3).weight = 1;
a.rule(3).connection = 1;
a.rule(4).antecedent = [3 2];
a.rule(4).consequent = [2];
a.rule(4).weight = 1;
a.rule(4).connection = 1
```

Yukarıdaki kod “run” edildiğinde;

a =

```
          name: 'riskAnaliz2'
          type: 'mamdani'
andMethod: 'min'
orMethod: 'max'
defuzzMethod: 'centroid'
impMethod: 'min'
aggMethod: 'max'
  input: [1x2 struct]
  output: [1x1 struct]
  rule: [1x4 struct]
```

Sonuçları gelir. Bunlar biz değiştirmedığımız sürece sistem tarafından varsayılan olarak kabul edilen değerlerdir.

>>fuzzy(a) ile arayüz görüntüleme yapılabilir.

>>showrule(a) ile kurallar görüntülenebilir.

Bazı çizim komutları:

>> plotfis(a)

>>plotmf(a,'input',1)

Kural listesi oluşturma işlemi;

Yukarıdaki kodlar için kural kısmı başta verilmeyip sonradan da verilebilir. Bunun için manuel olarak bir kural matrisi oluşturulmalıdır. Kural matrisi aşağıdaki ayarlara göre oluşturulur.

Dizinin her satırı, aşağıdaki biçimde bir kural içerir.

Sütun 1 - İlk giriş için üyelik fonksiyonu indisi

Sütun 2 - İkinci giriş için üyelik fonksiyonu indisi

Sütun 3 - Çıkış için üyelik fonksiyonu indisi

Sütun 4 - Kural ağırlığı

Sütun 5 - Bulanık operatör (AND için 1, OR için 2)

```
>>kurallar = [3 1 1 1 2;2 2 2 1 1;1 0 3 1 1;3 2 2 1 1];
```

```
>>a=addrule(a,kurallar);
```

```
>> showrule(a)
```

Üyelik fonksiyonlarının oluşturulması:

Üyelik fonksiyonları manuel olarak da oluşturulabilir. Bir m. file hazırlanırsa: RiskAnaliz2'nin ilk giriş değişkeninin ilk iki üyelik fonksiyonu.

```
clc; clear all ;close all;  
x=0:1:100;  
mf1 = trapmf(x, [-36 -4 25 47]);  
plot(x,mf1);  
hold on;  
mf2 = trimf(x, [28 52 76]);  
plot(x,mf2);  
ylim([-0.05 1.05]);  
deger1=trapmf(45, [-36 -4 25 47])  
deger2=trimf(45,[28 52 76] )
```

Yöntem 2. Daha az kod satırı ile gerçekleştirilir.

```
clear;clc;  
b = newfis('riskAnaliz3');  
b = addvar(b, 'input', 'ProjeButce', [0 100]);  
b = addmf(b, 'input', 1, 'yetersiz', 'trapmf', [-36 -4 25 47]);  
b = addmf(b, 'input', 1, 'sikisik', 'trimf', [28 52 76]);  
b = addmf(b, 'input', 1, 'yeterli', 'trapmf', [60 80 104 104]);  
b = addvar(b, 'input', 'CalisanSayisi', [0 100]);  
b = addmf(b, 'input', 2, 'az', 'trapmf', [-36 -4 30 65]);  
b = addmf(b, 'input', 2, 'fazla', 'trapmf', [35 69 100 100]);  
b = addvar(b, 'output', 'Risk', [0 100]);  
b = addmf(b, 'output', 1, 'dusuk', 'trapmf', [-36 -4 22 39]);  
b = addmf(b, 'output', 1, 'normal', 'trapmf', [30 46 54 70]);  
b = addmf(b, 'output', 1, 'yuksekk', 'trapmf', [61 78 100 100]);  
kurallar = [3 1 1 1 2;2 2 2 1 1;1 0 3 1 1;3 2 2 1 1];  
b = addrule(b,kurallar);
```

Durulanmış değerleri elde etme:

```
Workspace'de b var ise;  
>>z=evalfis(b,[65,50])%ProjeButce=%65, CalisanSayisi=%50 için çıkışı  
hesaplayan koddur.  
>>z=evalfis([65,50],b)  
b worksapce'de yok ise;  
>> b=readfis('RiskAnaliz2')
```

```
>>z=evalfis([65,50;30 67],b) % aynı anda önce (ProjeButce=%65,  
CalisanSayisi=%50) sonrasında (ProjeButce=%30, CalisanSayisi=%67) için  
çıktı üretir.
```

```
% Kod ile Bulanık Çıkarsama Sistemleri oluşturmanın avantajı gerçek zamanlı  
girdilere karşı gerçek zamanlı çıktılar oluşturabilmektir. Aynı anda 100  
adet giriş için 100 adet çıktı üretilebilir.
```

```
>> girisler=(0 + (100-0).*rand(100,2))% 100 satır 2 sütun giriş matrisi  
>> cikislar=evalfis(b,girisler);%100 adet çıkış oluşturuldu.
```